The Joint Block Diagonalization (JBD) problem: a tensor framework.

Dimitri Nion & Lieven De Lathauwer
K.U. Leuven, Kortrijk campus, Belgium
E-mails: Dimitri.Nion@kuleuven-kortrijk.be
Lieven.DeLathauwer@kuleuven-kortrijk.be

TDA 2010, Monopoli, Italy, September 13-17, 2010
Joint-Block-Diagonalization (JBD): model

\[ X_k = AD_kA^T \quad \text{or} \quad X_k = AD_kA^H \quad (+N_k), \quad k = 1, \ldots, K \]

JBD is a generalization of JD (Joint Diagonalization)/INDSCAL
JBD: ambiguities

Observation: if you choose $Z$ arbitrary, you lose the JBD structure.

Question: what is the structure of $Z$ such that the JBD model is still valid?
JBD: essential uniqueness

The JBD of \( \{X_k\}_{k=1}^K \) is said essentially unique if \( Z = \Lambda \Pi \)

- \( \Lambda \) an arbitrary block-diagonal matrix,
- \( \Pi \) an arbitrary block-wise permutation matrix.

\[ \begin{array}{ccc}
\Lambda_1 & \Lambda_2 & \Lambda_3 \\
\end{array} \]

Solving a JBD problem

Estimation of \( \{\text{Span}(A_r)\}_{r=1,\ldots,R} \) in an arbitrary order
JBD: State of the art

JBD is becoming popular signal processing tool in applications such as:

- Blind Source Separation (BSS) of convolutive mixtures in the time-domain,
- Independent Subspace Analysis.

Two approaches in the literature:

- **Approach 1:** Unitary-JBD [Abed Meraim and Belouchrani, 2004]
  « A is a square unitary matrix » \((A^T A = I)\)

- **Approach 2:** Non-Unitary JBD
  - **Approach 2.1:** [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]
    « A is tall and full column-rank. »
  → Indeed, their approach only works if A is a square non-unitary matrix.
  
- **Approach 2.2:** This talk
  « A can be a tall, square or fat non-unitary matrix »
  → JBD is a particular instance of Block-Component-Decompositions
  → Computation by a gradient-based algorithm
  → In the square case, better performance than 2.1
Joint-Block-Diagonalization : state of the art (1)

- **Approach 1**: Unitary-JBD [Abed Meraim and Belouchrani, 2004]
  - \( A \) is square unitary matrix \((A^T A = I)\)

\[
X_k = A D_k A^T (+ N_k) \quad \leftrightarrow \quad A^T X_k A = D_k + (A N_k A^T)
\]

\[
\max_A \sum_{k=1}^K \left\| \text{bdiag}(A^T X_k A) \right\|_F^2 \quad \text{or} \quad \min_A \sum_{k=1}^K \left\| \text{offbdiag}(A^T X_k A) \right\|_F^2
\]
Joint-Block-Diagonalization: state of the art (2)

  - $A$ is tall and full column-rank (Let $B = A^\dagger$, then $BA = I$)

\[
X_k = A D_k A^T (+ N_k) \quad \iff \quad BX_k B^T = D_k + (B N_k B^T)
\]

\[
\max_B \sum_{k=1}^{K} \left\| bdiag(BX_k B^T) \right\|_F^2 \quad \text{or} \quad \min_B \sum_{k=1}^{K} \left\| \text{offbdia}g(BX_k B^T) \right\|_F^2
\]

- 2 gradient-descent based algorithms, **JBD\textsubscript{OG}** and **JBD\textsubscript{ORG}** to solve

\[
\min_{\mathbf{B}} \phi_{\text{off}} = \sum_{k=1}^{K} \left\| \text{offbdiag}(\mathbf{B} \mathbf{X}_k \mathbf{B}^T) \right\|_F^2 \quad (1)
\]

**Drawbacks of approach 2:**

- \(\mathbf{B}=0\) is a trivial minimizer
- Under-determined case (\(\mathbf{A}\) fat, \(I<N\)) not handled, since it is assumed that \(\mathbf{B}\mathbf{A}=\mathbf{I}\)
- Indeed, the over-determined case (\(\mathbf{A}\) tall, \(I>N\)) is not successfully handled either because if \(\mathbf{B} = [\mathbf{B}_1^T, \mathbf{B}_2^T]^T = \mathbf{A}^\dagger\) is solution of an exact JBD problem, i.e.,

\[
\text{offbdiag} (\mathbf{B} \mathbf{X}_k \mathbf{B}^T) = \text{offbdiag} (\mathbf{B} \mathbf{A} \mathbf{D}_k \mathbf{A}^T \mathbf{B}^T) = \text{offbdiag} (\mathbf{D}_k) = 0, \; \forall k
\]

then \(\tilde{\mathbf{B}} = [(\mathbf{A}^\perp)^T, \mathbf{B}_2^T]^T\) is also solution of (1) but not solution of the JBD problem because

\[
\tilde{\mathbf{B}} \mathbf{A} = \begin{bmatrix} 0 & 0 \\ X & X \end{bmatrix}
\]

is not full rank. Idem for \(\tilde{\mathbf{B}} = [\mathbf{B}_1^T + (\mathbf{A}^\perp)^T, \mathbf{B}_2^T]^T\)
Joint-Block-Diagonalization: our contributions

- **Starting point:** the gradient-descent based algorithms $JBD_{OG}$ and $JBD_{ORG}$ of [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010] for Non-Unitary JBD can only handle the case where $A$ is square ($l=N$).

- **Main motivation:** Propose a novel technique to solve non-unitary JBD problems that can handle exactly-, over- and under- determined cases (i.e., $A$ may be square, tall or fat).

- **Main contributions:**
  - Formulate JBD as a tensor decomposition fitting problem;
  - Build a Conjugate Gradient (CG) algorithm to compute the tensor decomposition;
  - In the over-determined case, build a good starting point for any JBD algorithm;
  - Application: blind source separation via Independent Subspace Analysis (ISA).
JBD in tensor format: link to BCD

\[ \mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r \]

(or \( \mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r^* \))

JD \iff particular case of candecomp-parafac

JBD \iff particular case of BCD-(\(L_r, M_r, \_\)) terms

in case of hermitian symmetry)
**JBD in tensor format: conditions for essential uniqueness**

- **BCD-(L_r, M_r, ..):** \( \mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \mathord{\cdot}_1 A_r \mathord{\cdot}_2 B_r \) where \( A = [A_1, \ldots, A_R] \) is \( I \) by \( N \) \( B = [B_1, \ldots, B_R] \) is \( J \) by \( Q \)

- **Theorem [De Lathauwer, 2008]:** Suppose that \( \text{rank}(A) = N \), \( \text{rank}(B) = Q \), \( K > 2 \) and that the tensors \( \{\mathcal{D}_r\}_{r=1,\ldots,R} \) are generic, then the BCD-(L_r, M_r, ..) of \( \mathcal{X} \) is essentially unique (Sufficient condition).

- **JBD:** \( \mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \mathord{\cdot}_1 A_r \mathord{\cdot}_2 A_r \) where \( A = [A_1, \ldots, A_R] \) is \( I \) by \( N \)

- The same theorem can be invoked (the proof still holds with \( A \) instead of \( B \))

- In summary, it means that JBD is generically unique if

  \[
  K > 2 \quad \text{and} \quad \text{rank}(A) = N
  \]

- This is only a sufficient condition: uniqueness still holds but is harder to prove in several cases where the condition is not satisfied.

- For instance, uniqueness may still hold when \( \text{rank}(A) = I \) (fat, \( I < N \))
JBD in tensor format: cost function

\[
\min_{\{D_r\}_{r=1}^R, A} \phi_{LS} = \|\mathcal{H}\|_F^2 \quad \text{with} \quad \mathcal{H} = \mathcal{X} - \sum_{r=1}^R D_r \cdot_1 A_r \cdot_2 A_r
\]

\[
= \sum_{k=1}^K \|N_k\|_F^2 \quad \text{with} \quad N_k = X_k - \sum_{r=1}^R A_r D_{kr} A_r^T
\]

\[
= \|N\|_F^2 \quad \text{with} \quad N = X - (A \otimes A)D
\]

Where:
- \(A \otimes A = [A_1 \otimes A_1, \ldots, A_R \otimes A_R]\) is the Khatri-Rao product (block-wise Kronecker product).
- \(X\) and \(N\) are the \(I^2\times K\) matrix unfoldings of the \(I\times I\times K\) tensors \(\mathcal{X}\) and \(\mathcal{H}\), resp.
Real-valued data

\[ X_k = AD_k A^T + N_k \]

\[ \nabla_A (\phi_{LS}) = -2 \sum_{k=1}^{K} (N_k^T AD_k + N_k AD_k^T), \]

\[ \nabla_D (\phi_{LS}) = -2(A \otimes A^T)N \]

Complex-valued data, standard symmetry

\[ X_k = AD_k A^T + N_k \]

\[ \nabla_A^* (\phi_{LS}) = -\sum_{k=1}^{K} (N_k^T A^* D_k^* + N_k A^* D_k^H), \]

\[ \nabla_D^* (\phi_{LS}) = -(A \otimes A^*)^H N \]

Complex-valued data, hermitian symmetry

\[ X_k = AD_k A^H + N_k \]

\[ \nabla_A^* (\phi_{LS}) = -\sum_{k=1}^{K} (N_k^H AD_k + N_k AD_k^H), \]

\[ \nabla_D^* (\phi_{LS}) = -(A \otimes A^*)^T N \]
**Conjugate Gradient algorithm for JBD: JBD-CG**

\[ \text{while} \ (\text{stop criterion not satisfied}) \]

1. Compute \( \nabla_A \) and \( \nabla_D \) from \( D \) and \( A \)
2. Update search directions
   2.1. Compute \( \beta \) (e.g. with stabilized Polak - Ribière formula)
   2.2. New search directions:
   \[ d_A \leftarrow -\nabla_A + \beta d_A \]
   \[ d_D \leftarrow -\nabla_D + \beta d_D \]
3. Joint Exact Line Search:
   \[ (\alpha_A, \alpha_D) \leftarrow \arg \min_{\alpha_A, \alpha_D} \phi_{LS}(A + \alpha_A d_A, D + \alpha_D d_D) \]
4. Update unknowns:
   \[ A \leftarrow A + \alpha_A d_A \]
   \[ D \leftarrow D + \alpha_D d_D \]

*End*
Step 3 of JBD-CG : Joint Exact Line Search

\[ \min_{\alpha_A, \alpha_D} \phi_{LS}(A + \alpha_A d_A, D + \alpha_D d_D) \]

\[ \phi_{LS}(A + \alpha_A d_A, D + \alpha_D d_D) = \| [(A + \alpha_A d_A) \odot (A + \alpha_A d_A)](D + \alpha_D d_D) - X \|_F^2 \]

\[ = \alpha_D^2 Q_2(\alpha_A) + 2\alpha_D Q_1(\alpha_A) + Q_0(\alpha_A) \quad (1) \]

\[ \rightarrow \text{Solve } \frac{\partial \phi_{LS}(\alpha_A, \alpha_D)}{\partial \alpha_D} = 0 \quad \rightarrow \quad \alpha_D = \frac{-Q_1(\alpha_A)}{Q_2(\alpha_A)} \quad (2) \]

\[ \rightarrow \text{Substitute (2) in (1): } \phi_{LS}(\alpha_A) = \frac{-Q_1^2(\alpha_A) + Q_0(\alpha_A)Q_2(\alpha_A)}{Q_2(\alpha_A)} \quad (3) \]

\[ \rightarrow \text{Minimize (3) w.r.t. } \alpha_A \quad \text{(degree 7 polynomial rooting)} \]

\[ \rightarrow \text{Given } \alpha_A, \alpha_D \text{ is given by (2)} \]
JBD in tensor format: a closed form solution (when A is full column-rank)

**Idea:** when A is square or tall (full column-rank), the exact (noise-free) JBD problem can be solved by eigenvalue analysis. In presence of noise, this solution can be used to initialize iterative algorithms.

**Derivation in the square case (I=N):** Take two matrices

\[
\begin{align*}
X_1 &= AD_1A^T \\
X_2 &= AD_2A^T
\end{align*}
\]

We have \( X_{12} = X_1X_2^{-1} = AD_{12}A^{-1} \) where \( D_{12} \) is block-diagonal.

So \( X_{12}A_r = A_r[D_{12}]_{rr} \), i.e., \( \text{Span}(A_r) \), \( \forall r \), is an invariant subspace of \( X_{12} \).

**EVD:** \( X_{12} = EE^{-1} \), where \( E \) is diagonal.

We have \( A = EP \), where \( P \) is a permutation matrix that groups the eigenvectors \( L_r \) by \( L_r \).

Find \( P \) by checking the permuted block-diagonal structure of the matrices \( E^{-1}X_kE^{-T} = PD_kP^T, k = 1,...,K \)
Performance index

- We have: \( \hat{A} = A \Lambda \Pi + N \) where
  - \( \Lambda \) is an unknown block-diagonal matrix,
  - \( \Pi \) is an unknown block-wise permutation matrix,
  - \( N \) is the estimation error.

- Compute \( \Pi \) and \( \Lambda \) such that \( A \Lambda \Pi \) matches \( \hat{A} \) in the least squares sense.

- Compute the relative error:

\[
\varepsilon_{rel} = \frac{\| \hat{A} - A \Lambda \Pi \|_F}{\| A \|_F}
\]
Numerical experiments (1)

Exactly determined case: \( I=N=6, L_1=L_2=L_3=2, \ R=3, \ K=30 \)

Comparison of our JBD-CG technique with JBD-OG of \[ \text{[H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]} \]
Numerical experiments (2)

Under determined case: $I=6$, $N=8$, $L_1=L_2=L_3=L_4=2$, $R=4$, $K=100$
Application: Blind Subspace Separation

- **Primary sources** (mutually uncorrelated)
- Set of FIR filters
- **Secondary sources**:
  - Mutually correlated within the same subset
  - Mutually uncorrelated if not in the same subset

Linear instantaneous mixing (M sensors).

Mixing matrix:

$A$ is $M$ by $N$
Blind Subspace Separation via Second-Order-Statistics (SOS)

- \( y[p] = A x[p] \)
- Covariance matrix:
  \[
  R_{yy}[\tau] = E\{y[p]y^H[p + \tau]\} = A E\{x[p]x^H[p + \tau]\}A^H = ADA^H
  \]
  where \( D \) is block diagonal because on assumptions on secondary sources.
- Set of \( K \) covariance matrices:
  \[
  \begin{cases}
  R_{yy}[\tau_1] = AD_1A^H \\
  \vdots \\
  R_{yy}[\tau_K] = AD_KA^H
  \end{cases}
  \]
  JBD problem!
- Estimate \( A \) by JBD-ACG algorithm
- In the (over)-determined case (\( A \) square or tall), LS estimate of secondary sources:
  \[
  x[p] = A^{-1} y[p]
  \]
- Blind SIMO identification stage to recover primary sources from secondary ones,
Practical example: separation of speech signals

- R=3 primary sources.
- Number of filters for each source: \( L_1 = L_2 = L_3 = 4 \); each filter of length 50 generated randomly \( \rightarrow N=12 \) secondary sources.
- M=14 sensors (\( A \) is 14 by 12).
- K=21 covariance matrices.

\[ s_1 \quad s_2 \quad s_3 \]
Practical example: separation of speech signals

\[ \sum_{k=1}^{K} |R_{xx}[\tau_k]| = \sum_{k=1}^{K} |D_k| \]

\[ A^{-1} \hat{A} \]
Practical example: separation of speech signals

Separation is successful
Conclusion

- JBD is a generalization of JD
- JD $\leftrightarrow$ a particular case of Candecom/Parafac
  - JBD $\leftrightarrow$ a particular case of Block-Component-Decomposition (BCD)
- Uniqueness conditions for BCD can be invoked.
- In the exactly- and over- determined cases, we proposed an EVD-based technique useful for good initialization of JBD algorithms.
- We proposed a JBD-CG algorithm that works in exactly-, over- and under-determined cases
- Application: CG can accurately achieve blind subspace separation based on Second Order Statistics.